Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(8): 18295–18313.
DOI:10.3934/math.2023930
Received: 06 March 2023
Revised: 16 May 2023
Accepted: 23 May 2023
Published: 29 May 2023

Research article

A prediction-correction based proximal method for monotone variational inequalities with linear constraints

Feng Ma[1,*], **Bangjie Li**[1,*], **Zeyan Wang**[2], **Yaxiong Li**[1] **and Lefei Pan**[1]

[1] Xi’an Research institute of High-Tech, Xi’an 710025, China
[2] Department of Basic Education, Army Engineering University of PLA, Nanjing 211101, China

* **Correspondence:** Email: fengma@outlook.com, rosoldier@163.com.

Abstract: The monotone variational inequalities are being widely used as mathematical tools for studying optimal control problems and convex programming. In this paper, we propose a new prediction-correction method for monotone variational inequalities with linear constraints. The method consists of two procedures. The first procedure (prediction) utilizes projections to generate a predictor. The second procedure (correction) produces the new iteration via some minor computations. The main advantage of the method is that its main computational effort only depends on evaluating the resolvent mapping of the monotone operator, and its primal and dual step sizes can be enlarged. We prove the global convergence of the method. Numerical results are provided to demonstrate the efficiency of the method.

Keywords: variational inequalities; linear constraints; proximal algorithm; prediction-correction methods; contraction
Mathematics Subject Classification: 65K05, 65K10

1. Introduction

Let $\mathcal{D} \subset \Re^n$ be a nonempty, closed and convex set and $f : \Re^n \to \Re^n$ be a given continuous and monotone mapping. The variational inequality problem, denoted by VI$(\mathcal{D}, f)$, is to find a vector $x^* \in \mathcal{D}$ such that

$$\langle x - x^*, f(x^*)\rangle \geq 0, \quad \forall x \in \mathcal{D}. \tag{1.1}$$

The VI$(\mathcal{D}, f)$ has found many important applications in areas such as nonlinear complementarity problems (where $\mathcal{D} = \Re^n_+$) [1], traffic equilibrium and economic problems [2, 3]. For recent applications and numerical methods of the VI$(\mathcal{D}, f)$, we refer the reader to [4–6] and the references therein.

In this paper, we consider a special case of the general VI problem, where the set $\mathcal{D}$ is assumed to have the following form:

$$\mathcal{D} = \{x \in \Re^n | x \in \mathcal{X}, Ax = b\}, \tag{1.2}$$

here $A \in \Re^{m\times n}$ is a given matrix, $b \in \Re^m$ is a given vector, $\mathcal{X} \subset \Re^n$ is a nonempty, closed and convex subset. The solution set of (1.1) and (1.2), denoted by Ω^*, is assumed to be nonempty. Note that the VI problem (1.1) and (1.2) is closely related to the convex optimization problem with linear equality constraints:

$$\begin{array}{ll} \min & \theta(x) \\ \text{s.t.} & Ax = b, \ x \in \mathcal{X}. \end{array} \tag{1.3}$$

To show this, we recall the first order optimization conditions of problem (1.3). Let x^* be a minimum point of the convex function $\theta(x)$ over $\mathcal{D}$ and ξ^* be any vector in $\partial\theta(x^*)$, where $\partial\theta(\cdot)$ denotes the subdifferential operator of $\theta(\cdot)$. Then for any feasible direction $d \in \Re^n$ at x^*, we have $(\xi^*)^T d > 0$. It means that, the following variational inequality problem is captured:

$$\langle x - x^*, \xi^* \rangle \geq 0, \quad \forall x \in \mathcal{D}. \tag{1.4}$$

Thus, solving the VI problem (1.4) amounts to solving (1.3). This VI characterization is also used in e.g., [7,8].

The VI problem (1.1) and (1.2) or its equivalent form (1.3) is one of the fundamental problems in convex optimization. In particular, it includes linear programming, conic and semidefinite optimization as special cases. It can find many applications in compressed sensing, image processing and data mining, see, e.g., [9–11]. We refer to [12] for recent examples and discussions. To solve the VI problem (1.1) and (1.2), some proximal-like algorithms have been developed over the past years, see e.g., [13] for a review. One benchmark method is the augmented Lagrangian Method (ALM) [14,15] for nonlinear problems. The ALM is applicable to solve VI($\mathcal{D}, f$) (1.1) and (1.2). More specifically, for a given $u^k = (x^k, \lambda^k)$, ALM uses the following procedure to carry out the new iteration $u^{k+1} = (x^{k+1}, \lambda^{k+1}) \in \mathcal{X} \times \Re^m$:

Find $x^{k+1} \in \mathcal{X}$ such that

$$\langle x - x^{k+1}, f(x^{k+1}) - A^T\lambda^k + \beta A^T(Ax^{k+1} - b)\rangle \geq 0, \quad \forall x \in \mathcal{X}, \tag{1.5a}$$

then update λ^{k+1} via

$$\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b), \tag{1.5b}$$

where $\beta \geq 0$ is a given penalty parameter for the violation of the linearly constraints. To make ALM (1.5) more efficient and flexible, some alternative strategies can be used. For example, some self-adaptive rules can be carried to adjust the parameter β based on certain strategies [16–19]. We can also use some correction technologies to the output point [20,21]. Let $\tilde{u}^k$ denote the predictor generated by ALM. A simple and effective correction scheme is

$$u^{k+1} = u^k - \alpha_k(u^k - \tilde{u}^k), \tag{1.6}$$

where $0 < \alpha_k < 2$ is the step size parameter; see, e.g., [22,23] for details.

The main computational cost at each iteration of ALM is to solve the subproblem (1.5a), which is still a variational inequality, structurally the same as the original problem. So in many cases the

ALM (1.5a) is easy to iterate only if A is an identity matrix. This is because, for this case, the solution of the subproblem (1.5a) corresponds to the proximal mappings of f and it usually has closed-form solutions or can be efficiently solved up to a high precision. However, in many applications in sparse optimization, we often encounter the case where A is not an identity matrix, and the resulting subproblem (1.5a) no longer has closed-form solutions. For this case, solving the subproblem (1.5a) could be computationally intensive because of the costly evaluation of $(A^T A + \frac{1}{\beta} f)^{-1}(Av)$. Therefore, efficient numerical methods with implementable iterative scheme are highly desired.

Recently, several techniques attempting to overcome this difficulty have been proposed. In the framework of the proximal point algorithm (PPA), there are two relevant approaches. The first one is regularization. By adding a customized regularization term to the saddle-point reformulation of (1.1) and (1.2), the primal subproblem becomes easy as it only involves a simple evaluation, see the customized PPA algorithms proposed in e.g., [24–27]. We refer the reader to e.g., [28, 29] of the linearized regularization term for the separable case of problem (1.2). The second one employs prediction-correction technology which adds an asymmetrical proximal term to make a prediction, and then introduces a simple correction step to guarantee the convergence, see e.g., [30–32]. In this paper, we propose a new prediction-correction method for the VI$(\mathcal{D}, f)$. At each iteration, the method first makes a simple prediction step to obtain a point, and then generates a new iteration via a minor correction to the predictor. The reduced subproblems are easy to solve when the resolvent mapping of f can be efficiently evaluated. As can be seen in the Section 5, the proposed method converges faster with less iterations to achieve the same accuracy on most numerical cases.

The rest of this paper is organized as follows. In Section 2, we review some preliminaries which are useful for further analysis. In Section 1, we present the implementable prediction-correction method for VI$(\mathcal{D}, f)$. In Section 4, we establish the global convergence of the proposed method. The computational experiment is presented in Section 5. Finally, we make a conclusion in Section 6.

2. Preliminaries

2.1. Equivalent VI

In this section, we reformulate the VI problem (1.1) and (1.2) in succinct form. Let $\Omega = \mathcal{X} \times \Re^m$. By attaching a Lagrange multiplier vector $\lambda \in \Re^m$ to the linear constraints $Ax = b$, the VI problem (1.1) and (1.2) can be converted to the following form:

$$\left\langle \begin{array}{cc} x - x^*, & f(x^*) - A^T \lambda^* \\ \lambda - \lambda^*, & Ax^* - b \end{array} \right\rangle \geq 0, \quad \forall\, (x, \lambda)^T \in \Omega. \tag{2.1}$$

By denoting

$$u := \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) := \begin{pmatrix} f(x) - A^T \lambda \\ Ax - b \end{pmatrix}, \tag{2.2}$$

we can rewrite (2.1) into the following more compact form

$$\langle u - u^*, F(u^*) \rangle \geq 0, \quad \forall u \in \Omega. \tag{2.3}$$

Henceforth, we will denote the VI problem (2.2) and (2.3) by VI(Ω, F). Now, we make some basic assumptions and summarize some well known results of VI, which will be used in subsequent analysis.

2.2. Assumption

(A1) $\mathcal{X}$ is a simple closed convex set.

A set is said to be simple if the projection onto it can be easily obtained. Here, the projection of a point a onto the closed convex set $\mathcal{X}$, denoted by $P_{\mathcal{X}}(a)$, is defined as the nearest point $x \in \mathcal{X}$ to a, i.e.,

$$P_{\mathcal{X}}(a) = \operatorname{argmin}\{\|x - a\| \,|x \in \mathcal{X}\}.$$

(A2) The mapping f is point-to-point, monotone and continuous.

A mapping $F : \Re^n \to \Re^n$ is said to be monotone on Ω if

$$\langle u - v, F(u) - F(v)\rangle \geq 0, \quad \forall u, v \in \Omega. \tag{2.4}$$

(A3) The solution set of (2.2) and (2.3), denoted by Ω^*, is nonempty.

Remark 1. *The mapping $F(\cdot)$ defined in* (2.3) *is monotone with respect to Ω since*

$$(F(u) - F(\tilde{u}))^T(u - \tilde{u}) \geq 0, \;\; \forall u, \tilde{u} \in \Omega. \tag{2.5}$$

Proof.

$$\begin{aligned}(F(u) - F(\tilde{u}))^T(u - \tilde{u}) &= \begin{pmatrix} f(x) - f(\tilde{x}) - A^T(\lambda - \tilde{\lambda}) \\ A(x - \tilde{x}) \end{pmatrix}^T \begin{pmatrix} x - \tilde{x} \\ \lambda - \tilde{\lambda} \end{pmatrix} \\ &= (f(x) - f(\tilde{x}))^T (x - \tilde{x}) \\ &\geq 0, \end{aligned} \tag{2.6}$$

where the last inequality follows from the monotone property of f. □

2.3. Properties

Let G be a symmetric positive definite matrix. The G-norm of the vector u is denoted by $\|u\|_G := \sqrt{\langle u, Gu\rangle}$. In particular, when $G = I$, $\|u\| := \sqrt{\langle u, u\rangle}$ is the Euclidean norm of u. For a matrix A, $\|A\|$ denotes its norm $\|A\| := \max\{\|Ax\| : \|x\| \leq 1\}$.

The following well-known properties of the projection operator will be used in the coming analysis. The proofs can be found in textbooks, e.g., [2, 33].

Lemma 1. *Let $\mathcal{X} \in \Re^n$ be a nonempty closed convex set, $P_{\mathcal{X}}(\cdot)$ be the projection operator onto $\mathcal{X}$ under the G-norm. Then*

$$\langle y - P_{\mathcal{X}}(y), G(x - P_{\mathcal{X}}(y))\rangle \leq 0, \quad \forall y \in \Re^n, \forall x \in \mathcal{X}. \tag{2.7}$$

$$\|P_{\mathcal{X}}(x) - P_{\mathcal{X}}(y)\|_G \leq \|x - y\|_G, \quad \forall x, y \in \Re^n. \tag{2.8}$$

$$\|x - P_{\mathcal{X}}(y)\|_G^2 \leq \|x - y\|_G^2 - \|y - P_{\mathcal{X}}(y)\|_G^2, \quad \forall y \in \Re^n, \forall x \in \mathcal{X}. \tag{2.9}$$

For any arbitrary positive scalar β and $u \in \Omega$, let $e(u,\beta)$ denote the residual function associated with the mapping F, i.e.,

$$e(u,\beta) = u - P_{\Omega}[u - \beta F(u)]. \tag{2.10}$$

Lemma 2. *Solving VI(Ω, F) is equivalent to finding the zero point of the mapping*

$$e(u,\beta) := \begin{pmatrix} e_1(u,\beta) \\ e_2(u,\beta) \end{pmatrix} = \begin{pmatrix} x - P_{\mathcal{X}}\{x - \beta[f(x) - A^T\lambda]\} \\ \beta(Ax - b) \end{pmatrix}. \tag{2.11}$$

Proof. See [2], pp 267. □

3. The prediction-correction method

We now formally present the new prediction-correction method for the VI(Ω, F) (Algorithm 1). The method can be viewed as a generalization of [31] in relaxed case.

Algorithm 1: A prediction-correction based method (PCM) for the VI(Ω, F).

Step 0: Initialization step.
Given a small number $\epsilon > 0$. Take $\gamma \in (0, 2)$, $u^0 \in \Re^{n+m}$; set $k = 0$. Choose the parameters $r > 0, s > 0$, such that

$$rs > \frac{1}{4}\|A^TA\|. \tag{3.1}$$

Step 1: Prediction step.
Generate the predictor $\tilde{x}^k$ via solving the following projection equation:

$$\tilde{x}^k = P_{\mathcal{X}}[x^k - \frac{1}{r}(f(\tilde{x}^k) - A^T\lambda^k)]. \tag{3.2a}$$

Then update $\tilde{\lambda}^k \in \Re^m$ via

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b). \tag{3.2b}$$

Step 2: Correction step.
Generate the new iteration $u^{k+1} = (x^{k+1}, \lambda^{k+1})$ via

$$x^{k+1} = x^k - \gamma(x^k - \tilde{x}^k) - \frac{\gamma}{2r}A^T(\lambda^k - \tilde{\lambda}^k), \tag{3.3a}$$

and

$$\lambda^{k+1} = \lambda^k + \frac{\gamma}{2s}A(x^k - \tilde{x}^k) - \gamma(I - \frac{1}{2rs}AA^T)(\lambda^k - \tilde{\lambda}^k). \tag{3.3b}$$

Step 3: Convergence verification.
If $\|u^{k+1} - u^k\| \le \epsilon$, stop; otherwise set $k := k + 1$; go to Step 1.

Remark 2. *Note that the regularized projection Eq (3.2a) amounts to solving the following VI problem*

$$\langle x - \tilde{x}^k, \frac{1}{r}(f(\tilde{x}^k) - A^T\lambda^k) + (\tilde{x}^k - x^k)\rangle \ge 0, \quad \forall x \in \mathcal{X}, \tag{3.4}$$

which represents to find $\tilde{x}^k \in \mathcal{X}$ such that

$$0 \in \tilde{x}^k + \frac{1}{r} f(\tilde{x}^k) - (\frac{1}{r} A^T \lambda^k + x^k). \tag{3.5}$$

We can rewrite the above equation as

$$\tilde{x}^k \in (I + \frac{1}{r} f)^{-1} (\frac{1}{r} A^T \lambda^k + x^k). \tag{3.6}$$

Thus, the subproblem (3.2a) is equivalent to an evaluation of the resolvent operator $(I + \frac{1}{r} f)^{-1}$ when $\mathcal{X}$ is the finite-dimensional Euclidean spaces [34]. Notice that ALM (1.5a) needs to evaluate the operator $(A^T A + \frac{1}{\beta} f)^{-1}$. Therefore, the resulting subproblem (3.2a) could be much easier to solve than ALM (1.5a). On the other hand, the correction step only consists of some elementary manipulations. Thus, the resulting method (3.2a)–(3.3b) *is easily implementable.*

Remark 3. *The parameters $\frac{1}{r}$ and $\frac{1}{s}$ in the prediction step can be viewed as two step sizes for the projection step* (3.2a) *and dual step, respectively. Using the step size condition $rs > \frac{1}{4}\|A^T A\|$ of this algorithm, the parameters $\frac{1}{r}$ and $\frac{1}{s}$ can be chosen larger values compared to the condition $rs > \|A^T A\|$ of some other primal dual algorithms, e.g., linearized ALM, customized PPA algorithms. This larger step size is usually beneficial to the effectiveness and robustness of the algorithm. In Section 5 we will empirically see that our algorithm is significantly faster than some other primal dual algorithms by allowing larger step sizes.*

4. Convergence properties

In the following, we will focus our attention to solving VI(Ω, F). But at the beginning, to make the notation of the proof more succinct, we define some matrices:

$$G = \begin{pmatrix} rI & \frac{1}{2}A^T \\ \frac{1}{2}A & sI \end{pmatrix}, \quad Q = \begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix}. \tag{4.1}$$

Obviously, when $rs > \frac{1}{4}\|A^T A\|$, G is a positive definite matrix. Now, we start to prove the global convergence of the sequence $\{u^k\}$. Towards this end, we here follow the work [35] to reformulate the algorithm into a prediction-correction method and establish its convergence results. We first prove some lemmas. The first lemma quantifies the discrepancy between the point $\tilde{u}^k$ and a solution point of VI(Ω, F).

Lemma 3. *Let $\{u^k\}$ be generated by the PCM and $\{\tilde{u}^k\}$ be generated by PCM* (3.2)*, and the matrix Q be given in (4.1). Then we have*

$$\langle u - \tilde{u}^k, F(\tilde{u}^k) - Q(u^k - \tilde{u}^k)\rangle \geq 0, \quad \forall u \in \Omega. \tag{4.2}$$

Proof. Note that the sequence $\{\tilde{u}^k\}$ generated by (3.2) is actually solutions of the following VIs:

$$\langle x - \tilde{x}^k, f(\tilde{x}^k) - A^T \lambda^k - r(x^k - \tilde{x}^k)\rangle \geq 0, \quad \forall x \in \mathcal{X}, \tag{4.3}$$

and

$$\langle \lambda - \tilde{\lambda}^k, A\tilde{x}^k - b - s(\lambda^k - \tilde{\lambda}^k)\rangle \geq 0, \quad \forall \lambda \in \Re^m. \tag{4.4}$$

Combining (4.3) and (4.4) yields

$$\left\langle \begin{array}{cc} x - \tilde{x}^k, & f(\tilde{x}^k) - A^T\tilde{\lambda}^k - A^T(\lambda^k - \tilde{\lambda}^k) - r(x^k - \tilde{x}^k) \\ \lambda - \tilde{\lambda}, & A\tilde{x}^k - b - s(\lambda^k - \tilde{\lambda}^k) \end{array} \right\rangle \geq 0, \quad \forall u \in \Omega. \tag{4.5}$$

It can be rewritten as

$$\left\langle \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}, \begin{pmatrix} f(\tilde{x}^k) - A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} - \begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \right\rangle \geq 0, \quad \forall u \in \Omega. \tag{4.6}$$

Using the notation of $F(u)$ (2.3) and Q (4.1), the assertion (4.2) is proved. □

The following lemma characterizes the correction step by a matrix form.

Lemma 4. *Let* $\{u^k\}$ *be generated by the PCM and* $\{\tilde{u}^k\}$ *be generated by PCM* (3.2)*, Then we have*

$$u^k - u^{k+1} = \gamma M(u^k - \tilde{u}^k), \tag{4.7}$$

where

$$M = \begin{pmatrix} I & \frac{1}{2r}A^T \\ -\frac{1}{2s}A & I - \frac{AA^T}{2rs} \end{pmatrix}. \tag{4.8}$$

Proof. From the correction Step (3.3a) and (3.3b), we have

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} (x^k - \tilde{x}^k) + \frac{A^T}{2r}(\lambda^k - \tilde{\lambda}^k) \\ -\frac{1}{2s}A(x^k - \tilde{x}^k) + (I - \frac{1}{2rs}AA^T)(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}, \tag{4.9}$$

which can be written as

$$\begin{pmatrix} x^k - x^{k+1} \\ \lambda^k - \lambda^{k+1} \end{pmatrix} = \gamma \begin{pmatrix} I & \frac{1}{2r}A^T \\ -\frac{1}{2s}A & I - \frac{AA^T}{2rs} \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{4.10}$$

By noting the matrix M (4.8), the proof is completed. □

Using the matrices Q (4.1) and M (4.8), we define

$$H = QM^{-1}, \tag{4.11}$$

then we have $Q = HM$. The inequality (4.2) can be written as

$$\gamma\langle u - \tilde{u}^k, F(\tilde{u}^k) - HM(u^k - \tilde{u}^k)\rangle \geq 0, \quad \forall u \in \Omega. \tag{4.12}$$

Substituting (4.7) into (4.12) and using the monotonicity of F (see (2.5)), we obtain

$$\tilde{u}^k \in \Omega, \quad \langle u - \tilde{u}^k, F(u)\rangle \geq \langle u - \tilde{u}^k, H(u^k - u^{k+1})\rangle, \quad \forall u \in \Omega. \tag{4.13}$$

Now, we prove a simple fact for the matrix H in the following lemma.

Lemma 5. *The matrix* H *defined in* (4.11) *is positive definite for any* $r > 0, s > 0$*,* $rs > \frac{1}{4}\|A^TA\|$.

Proof. For the matrix Q defined by (4.1), we have

$$Q^{-1} = \begin{pmatrix} \frac{1}{r}I & -\frac{1}{rs}A^T \\ 0 & \frac{1}{s}I \end{pmatrix}.$$

Thus, it follows from (4.1) that

$$\begin{aligned} H^{-1} &= MQ^{-1} = \begin{pmatrix} I & \frac{1}{2r}A^T \\ -\frac{1}{2s}A & I - \frac{AA^T}{2rs} \end{pmatrix} \begin{pmatrix} \frac{1}{r}I & -\frac{1}{rs}A^T \\ 0 & \frac{1}{s}I \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{r}I & -\frac{1}{2rs}A^T \\ -\frac{1}{2rs}A & \frac{1}{s}I \end{pmatrix}. \end{aligned} \tag{4.14}$$

For any any $r > 0, s > 0$ satisfying $rs > \frac{1}{4}\|A^TA\|$, H^{-1} is positive definite, and the positive definiteness of H is followed directly. □

Then we show how to deal with the right-hand side of (4.13). The following lemma gives an equivalent expression of it in terms of $\|u - u^k\|_H$ and $\|u - u^{k+1}\|_H$.

Lemma 6. *Let* $\{u^k\}$ *be generated by the PCM and* $\{\tilde{u}^k\}$ *be generated by PCM* (3.2). *Then we have*

$$\langle u - \tilde{u}^k, H(u^k - u^{k+1})\rangle = \frac{1}{2}(\|u - u^{k+1}\|_H^2 - \|u - u^k\|_H^2) + \frac{1}{2}\|u^k - \tilde{u}^k\|_G^2, \ \forall u \in \Omega, \tag{4.15}$$

where the matrix $G = \gamma(Q^T + Q) - \gamma^2 M^T HM$.

Proof. Applying the identity

$$\langle a - b, H(c - d)\rangle = \frac{1}{2}(\|a - d\|_H^2 - \|a - c\|_H^2) + \frac{1}{2}(\|c - b\|_H^2 - \|d - b\|_H^2), \tag{4.16}$$

to the right term of (4.13) with $a = u, b = \tilde{u}^k, c = u^k, d = u^{k+1}$, we obtain

$$\langle u - \tilde{u}^k, H(u^k - u^{k+1})\rangle \ = \ \frac{1}{2}(\|u - u^{k+1}\|_H^2 - \|u - u^k\|_H^2) + \frac{1}{2}(\|u^k - \tilde{u}^k\|_H^2 - \|u^{k+1} - \tilde{u}^k\|_H^2). \tag{4.17}$$

For the last term of (4.17), we have

$$\begin{aligned} &\|u^k - \tilde{u}^k\|_H^2 - \|u^{k+1} - \tilde{u}^k\|_H^2 \\ &\quad = \|u^k - \tilde{u}^k\|_H^2 - \|(u^k - \tilde{u}^k) - (u^k - u^{k+1})\|_H^2 \\ &\quad \overset{(4.7)}{=} \|u^k - \tilde{u}^k\|_H^2 - \|(u^k - \tilde{u}^k) - \gamma M(u^k - \tilde{u}^k)\|_H^2 \\ &\quad = 2\gamma\langle u^k - \tilde{u}^k, HM(u^k - \tilde{u}^k)\rangle - \gamma^2\langle u^k - \tilde{u}^k, M^THM(u^k - \tilde{u}^k)\rangle \\ &\quad \overset{(4.11)}{=} \langle u^k - \tilde{u}^k, (\gamma(Q^T + Q) - \gamma^2 M^THM)(u^k - \tilde{u}^k)\rangle. \end{aligned} \tag{4.18}$$

the assertion is proved. □

Now, we investigate the positive definiteness for the matrix G. Using (4.11), we have

$$G \ = \ \gamma(Q^T + Q) - \gamma^2 M^T HM$$

$$
\begin{aligned}
&= \gamma(Q^T + Q) - \gamma^2 M^T Q \\
&= \gamma\begin{pmatrix} 2rI & A^T \\ A & 2sI \end{pmatrix} - \gamma^2 \begin{pmatrix} I & -\frac{1}{2s}A^T \\ \frac{1}{2r}A & I - \frac{AA^T}{2rs} \end{pmatrix}\begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix} \\
&= \gamma\begin{pmatrix} 2rI & A^T \\ A & 2sI \end{pmatrix} - \gamma^2 \begin{pmatrix} rI & \frac{1}{2}A^T \\ \frac{1}{2}A & sI \end{pmatrix} \\
&= (2\gamma - \gamma^2)\begin{pmatrix} rI & \frac{1}{2}A^T \\ \frac{1}{2}A & sI \end{pmatrix}. \qquad (4.19)
\end{aligned}
$$

Thus, when $rs > \frac{1}{4}\|A^TA\|$ and $\gamma \in (0, 2)$, the matrix G is guaranteed to be positive definite, and we can easily obtained the contraction property of the algorithm. This is given by the following theorem.

Theorem 1. *Suppose the condition*

$$rs > \frac{1}{4}\|A^TA\| \qquad (4.20)$$

holds. Let the relaxation factor $\gamma \in (0, 2)$. Then, for any $u^* = (x^*, \lambda^*)^T \in \Omega^*$, the sequence $u^{k+1} = (x^{k+1}, \lambda^{k+1})^T$ generated by PCM satisfies the following inequality:

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - \|u^k - \tilde{u}^k\|_G^2. \qquad (4.21)$$

Proof. Combining (4.13) and (4.15), we have

$$
\begin{aligned}
&\tilde{u}^k \in \Omega, \quad \langle u - \tilde{u}^k, F(u)\rangle \\
&\ge \frac{1}{2}(\|u - u^{k+1}\|_H^2 - \|u - u^k\|_H^2) + \frac{1}{2}\|u^k - \tilde{u}^k\|_G^2, \quad \forall u \in \Omega. \qquad (4.22)
\end{aligned}
$$

Note that $u^* \in \Omega$. We get

$$\|u^k - u^*\|_H^2 - \|u^{k+1} - u^*\|_H^2 \ge \|u^k - \tilde{u}^k\|_G^2 + 2\langle \tilde{u}^k - u^*, F(u^*)\rangle.$$

Since u^* is a solution of (2.3) and $\tilde{u}^k \in \Omega$, we have

$$\langle \tilde{u}^k - u^*, F(u^*)\rangle \ge 0, \qquad (4.23)$$

and thus

$$\|u^k - u^*\|_H^2 - \|u^{k+1} - u^*\|_H^2 \ge \|u^k - \tilde{u}^k\|_G^2.$$

The assertion (4.21) follows directly. □

Based on the above results, we are now ready to prove the global convergence of the algorithm.

Theorem 2. *Let $\{u^k\}$ be the sequence generated by PCM for the VI problem* (2.2) *and* (2.3)*. Then, for any $r > 0, s > 0$ satisfying $rs > \frac{1}{4}\|A^TA\|$ and $\gamma \in (0, 2)$, the sequence $\{u^k\}$ converges to a solution of VI(Ω, F).*

Proof. We follows the standard analytic framework of contraction-type methods to prove the convergence of the proposed algorithm. It follows from Theorem 1 that $\{u^k\}$ is bounded. Then we have that

$$\lim_{k\to\infty} \|u^k - \tilde{u}^k\|_G = 0. \qquad (4.24)$$

Consequently,

$$\lim_{k\to\infty} \|x^k - \tilde{x}^k\| = 0, \tag{4.25}$$

and

$$\lim_{k\to\infty} \|A\tilde{x}^k - b\| = \lim_{k\to\infty} \|s(\lambda^k - \tilde{\lambda}^k)\| = 0. \tag{4.26}$$

Since

$$\tilde{x}^k = P_X[\tilde{x}^k - \frac{1}{r}(f(\tilde{x}^k) - A^T\tilde{\lambda}^k) + (x^k - \tilde{x}^k) + \frac{1}{r}A^T(\lambda^k - \tilde{\lambda}^k)], \tag{4.27}$$

and

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b), \tag{4.28}$$

it follows from (4.25) and (4.26) that

$$\begin{cases} \lim\limits_{k\to\infty} \tilde{x}^k - P_X[\tilde{x}^k - \frac{1}{r}(f(\tilde{x}^k) - A^T\tilde{\lambda}^k)] = 0, & (4.29a) \\ \lim\limits_{k\to\infty} A\tilde{x}^k - b = 0. & (4.29b) \end{cases}$$

Because $\tilde{u}^k$ is also bounded, it has at least one cluster point. Let u^∞ be a cluster point of $\tilde{u}^k$ and let $\tilde{u}^k_j$ be the subsequence converges to u^∞. It follows from (4.29) that

$$\begin{cases} \lim\limits_{j\to\infty} \tilde{x}^k_j - P_X[\tilde{x}^k_j - \frac{1}{r}(f(\tilde{x}^k_j) - A^T\tilde{\lambda}^k_j)] = 0, & (4.30a) \\ \lim\limits_{j\to\infty} A\tilde{x}^k_j - b = 0. & (4.30b) \end{cases}$$

Consequently, we have

$$\begin{cases} x^\infty - P_X[x^\infty - \frac{1}{r}(f(x^\infty) - A^T\lambda^\infty)] = 0, & (4.31a) \\ Ax^\infty - b = 0. & (4.31b) \end{cases}$$

Using the continuity of F and the projection operator $P_\Omega(\cdot)$, we have that u^∞ is a solution of VI(Ω, F). On the other hand, by taking limits over the subsequences in (4.24) and using $\lim_{j\to\infty} \tilde{u}^k_j = u^\infty$. we have that, for any $k > k_j$,

$$\|u^k - u^\infty\|_H \leq \|u^{k_j} - u^\infty\|_H.$$

Thus, the sequence $\{u^k\}$ converges to u^∞, which is a solution of VI(Ω, F). □

5. Numerical experiments

In this paper, we test the performance of PCM (3.2a)–(3.3b) for solving the basis pursuit (BP) and matrix completion problem. All the simulations are performed on a Lenovo Laptop with CPU Intel with 2.81GHz and 16GB RAM memory, using Matlab R2015b.

5.1. Basis pursuit problem

The BP problem arises from areas such as the communities of information theory, signal processing, statistics, machine learning. it seeks to encode a large sparse signal representations through a relatively small number of linear equations. The BP problem can be cast as the following equality-constrained l_1 minimization problem

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = b, \end{aligned} \tag{5.1}$$

where $x \in \Re^n$, data $A \in \Re^{m\times n}$ with $m < n$, $b \in \Re^m$. Here we assume that A has full row rank. By invoking the first-order optimality condition, BP is equivalent to VI (1.1) and (1.2) with $f(x) = \partial\|x\|_1$. Applying PCM with $\gamma = 1$ for this problem, we get the following iterative scheme:

$$\begin{cases} \tilde{x}^k \in P_{\Re^n}[x^k - \frac{1}{r}(\partial\|\tilde{x}^k\|_1 - A^T\lambda^k)], & \text{(5.2a)} \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b), & \text{(5.2b)} \\ x^{k+1} = \tilde{x}^k - \frac{1}{2r}A^T(\lambda^k - \tilde{\lambda}^k), & \text{(5.2c)} \\ \lambda^{k+1} = \lambda^k + \frac{1}{2s}A(x^k - \tilde{x}^k) - (I - \frac{AA^T}{2rs})(\lambda^k - \tilde{\lambda}^k). & \text{(5.2d)} \end{cases}$$

Note that the projection (5.2a) is equivalent to the l_1 shrinkage operation:

$$\tilde{x}^k := \text{Shrink}(x^k + \frac{1}{r}A^T\lambda^k, \frac{1}{r}),$$

where the l_1 shrinkage operator, denoted by Shrink(M, ξ), is defined as

$$[\text{Shrink}(M,\xi)]_i := \begin{cases} M_i - \xi, & \text{if } M_i > \xi, \\ M_i + \xi, & \text{if } M_i < -\xi, \qquad i = 1, 2, \ldots, n. \\ 0, & \text{if } |M_i| \le \xi, \end{cases}$$

In our experiments, we focus on comparing our algorithm with the linearized ALM [36] (L-ALM) and the customized PPA (C-PPA) in [27] and verifying its efficiency. Similar with PCM, L-ALM and C-PPA also depend on Shrink operation, which have the same easiness of implementation.

The data used in this experiment is similar to the one employed in [37]. The basic setup of the problem is as follows. The data matrix A is a i.i.d. standard Gaussian matrix generated by the randn(m, n) function in MATLAB with $m = n/2$. The original sparse signal x_{true} is sampled from i.i.d. standard Gaussian distributions with $m/5$ nonzero values. The output b is then created as the signs of $b = Ax$. It is desirable to test problems that have a precisely known solution. In fact, when the original signal x_{true} is very sparse, it reduces to a minimization problem. The parameters used in the numerical experiments are similar to that in [19, 38]: we set the relaxation factor $\gamma = 1$, $s \in (25, 100)$, $r = 1.01\|A^TA\|/(4s)$ for PCA and $s \in (25, 100)$, $r = 1.01\|A^TA\|/s$ for C-PPA. (In order to ensure the convergence of L-ALM and C-PPA, the parameters r, s should satisfy $rs > \|A^TA\|$.) We set the criterion error as $\min\{\text{rel}_L = \|x^k - x^{k-1}\|, \text{rel}_S = Ax^k - b\}$, and declare successful recovery when this error is less than $Tol = 10^{-3}$. In all the tests, the initial iteration is $(x^0, \lambda^0) = (\mathbf{0}, \mathbf{1})$.

We first test the sensitivity of γ of PCM. We fix $s = 50$ and choose different values of γ in the interval $[0.6, 1.8]$. The number of iterations required are reported in Figure 1. The curves in Figure 1 indicate that $\gamma \in (1, 1.4)$ is preferable when we implement Algorithm PCM in practice.

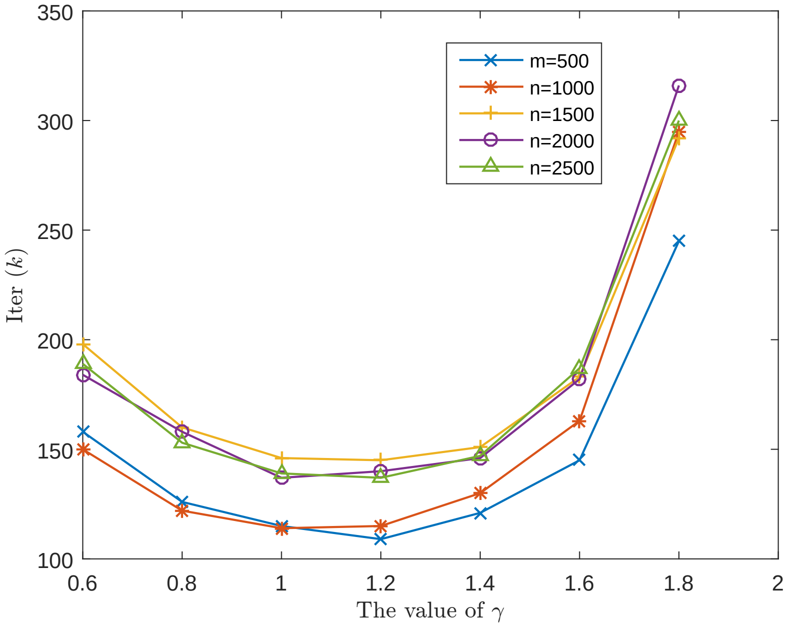

Figure 1. Sensitivity test on the relaxation factor of γ.

In order to investigate the stability and efficiency of our algorithms, we test 8 groups of problems with different n and we generated the model by 10 times and reported the average results. The comparisons of these algorithms for small BP problems are presented in Tables 1–3.

Table 1. Numerical Results for Basis Pursuit problem $s = 25$.

	PCM				L-ALM				C-PPA			
n	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)
100	81	4.2e-04	1.9e-04	0.03	161	1.6e-04	7.5e-04	0.03	232	8.2e-05	8.6e-04	0.04
300	88	5.1e-04	7.3e-04	0.05	207	7.5e-05	6.3e-04	0.07	311	5.3e-05	9.4e-04	0.10
600	110	9.2e-05	7.9e-04	0.54	262	7.2e-05	8.9e-04	0.62	374	6.0e-05	9.4e-04	1.01
1000	141	4.4e-05	9.8e-04	1.81	291	3.3e-05	7.9e-04	2.21	411	3.2e-05	9.4e-04	2.86
1500	139	6.7e-05	6.6e-04	3.95	359	7.0e-05	9.4e-04	5.04	484	6.9e-05	9.0e-04	6.54
2000	151	6.3e-05	7.9e-04	6.55	406	1.2e-05	9.7e-04	9.24	536	1.3e-05	9.7e-04	12.08
2500	171	3.9e-05	8.5e-04	11.57	519	1.3e-05	8.9e-04	18.60	635	1.8e-05	9.8e-04	22.69
3000	199	2.9e-05	8.8e-04	18.82	585	1.7e-05	9.8e-04	29.46	709	1.7e-05	9.0e-04	35.61

Table 2. Numerical Results for Basis Pursuit problem $s = 50$.

	PCM				L-ALM				C-PPA			
n	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)
100	80	5.9e-04	7.0e-04	0.03	151	3.7e-04	9.6e-04	0.03	224	1.4e-04	9.4e-04	0.04
300	87	6.1e-04	8.1e-04	0.05	203	1.2e-04	8.5e-04	0.09	299	3.3e-05	9.0e-04	0.10
600	110	1.1e-04	6.1e-04	0.49	249	3.5e-05	9.9e-04	0.63	331	5.7e-05	8.6e-04	0.87
1000	137	4.0e-05	8.5e-04	1.75	266	3.2e-05	7.9e-04	1.80	370	3.1e-05	8.3e-04	2.62
1500	135	3.5e-05	8.6e-04	3.54	312	3.6e-05	8.5e-04	4.35	424	3.4e-05	8.2e-04	5.84
2000	146	5.4e-05	7.6e-04	6.38	331	1.3e-05	9.8e-04	7.57	445	1.5e-05	9.5e-04	10.19
2500	143	4.1e-05	9.7e-04	9.67	375	1.4e-05	9.3e-04	13.41	491	1.5e-05	9.1e-04	17.63
3000	170	3.1e-05	9.7e-04	16.00	418	1.5e-05	9.5e-04	20.70	533	1.6e-05	8.9e-04	26.42

Table 3. Numerical Results for Basis Pursuit problem $s = 75$.

	PCM				L-ALM				C-PPA			
n	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)	Iter.	rel$_L$	rel$_S$	CPU(s)
100	82	5.3e-04	5.7e-04	0.02	164	2.7e-04	6.2e-04	0.03	236	6.3e-05	9.2e-04	0.04
300	99	1.9e-04	7.9e-04	0.05	213	9.4e-05	5.9e-04	0.07	284	4.2e-05	9.2e-04	0.09
600	117	6.7e-05	8.9e-04	0.55	226	1.1e-04	9.9e-04	0.58	316	8.5e-05	9.3e-04	0.84
1000	179	4.2e-05	8.7e-04	2.35	253	5.4e-05	9.5e-04	1.68	369	1.9e-05	9.5e-04	2.47
1500	159	3.4e-05	9.3e-04	4.20	299	3.4e-05	8.1e-04	4.13	399	3.5e-05	7.9e-04	5.39
2000	142	6.6e-05	9.4e-04	6.18	313	1.3e-05	9.0e-04	7.32	410	1.9e-05	9.7e-04	9.56
2500	139	2.9e-05	9.8e-04	9.41	337	1.4e-05	1.0e-03	12.20	444	1.1e-05	9.8e-04	16.00
3000	168	3.7e-05	9.0e-04	15.92	370	1.6e-05	8.2e-04	18.54	464	2.2e-05	9.0e-04	23.22

From Tables 1–3, it can be seen that the PCM performs the best, both in terms of number of iterations and CPU time for all test cases. These numerical results illustrate that if the step size condition relaxed, can indeed be beneficial to yield larger step sizes, which could accelerate the convergence of algorithm.

To verify the performance results of our algorithm, we plotted the approximation error $\text{rel}_L = \|x^k - x^{k-1}\|, \text{rel}_S = \|Ax^k - b\|$ achieved for $n = 1500, s = 50$ by each of the algorithms versus the iteration number k in Figures 2 and 3, respectively. It is clear to see that PCM outperforms all the other algorithms significantly in terms of number of iterations.

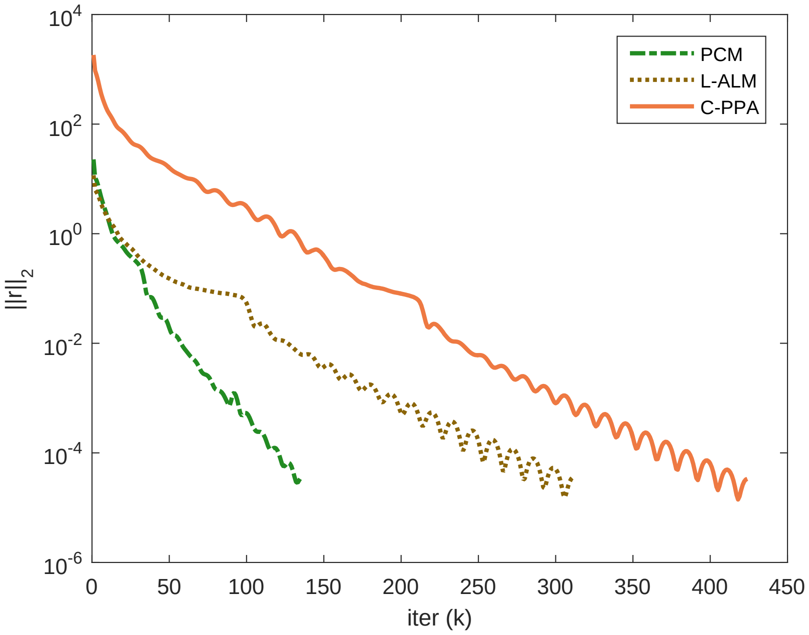

Figure 2. Relative errors of $\underline{\mathrm{rel}_L} = \|x^k - x^{k-1}\|$.

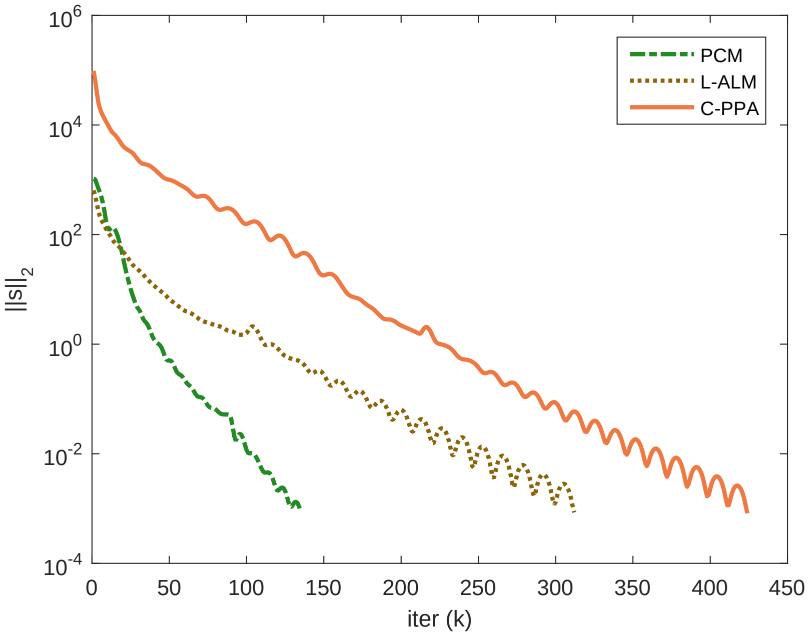

Figure 3. Relative errors of $\underline{\mathrm{rel}_S} = \|Ax^k - b\|$.

5.2. Matrix completion problem

Matrix completion problem (MC) comes from many fields such as signal processing, statistics, machine learning communities. It tries to recover the low-rank matrix X from its incomplete known entries. Mathematically, its convex formula is as follows:

$$\begin{aligned} \min \quad & \|X\|_* \\ \text{s.t.} \quad & X_{ij} = M_{ij}, (i, j) \in \Omega, \end{aligned} \tag{5.3}$$

where $\|X\|_*$ is the nuclear norm of X, M is the unknown matrix with p available sampled entries and Ω is a set of pairs of indices of cardinality p. By invoking the first-order optimality condition, MC can also be equivalent to VI (1.1) and (1.2) with $f(x) = \partial\|X\|_*$.

The basic setup of the problem is as follows. We first generate two random matrices $M_L \in \Re^{n\times r_a}$ and $M_R \in \Re^{n\times r_a}$, all with i.i.d. standard Gaussian entries, and then set the low-rank matrix $M = M_L M_R^T$. The available index set Ω is randomly uniformly sampled in all cardinality sets $|\Omega|$. We denote the oversampling factor (OS) by $OS = |\Omega|/r_a(2n - r_a)$, i.e., the ratio of sample size to degrees of freedom in an asymmetric matrix of rank r_a. The relative error of the approximation X is defined as

$$\text{relative error} = \frac{\|X_\Omega - M_\Omega\|_F}{\|M_\Omega\|_F}. \tag{5.4}$$

We set the relative error $Tol = 10^{-5}$ as the tolerance for all algorithms. In all tests, the initial iteration is $(X^0, \Lambda^0) = (\mathbf{0}, \mathbf{0})$. The parameters used in the numerical experiments are set as follows: we set $r = 0.006$, $s = 1.01/(4r)$, $\gamma = 1$ for PCM, $s = 1.01/r$ for L-ALM and C-PPA.

Tables 4–6 list the comparison between these algorithms with three different OS values. The results confirm that PCM outperforms other methods in terms of computation time and number of iterations in all cases.

Table 4. Comparison results of PCM, L-ALM, C-PPA ($OS = 5$).

Problems		PCM		L-ALM		C-PPA	
n	r_a	Rel.err.	Iter.	Rel.err.	Iter.	Rel.err.	Iter.
100	5	9.71e-06	60	6.22e-05	101	5.83e-05	101
100	10	9.74e-06	18	9.17e-06	38	9.15e-06	37
200	5	9.40e-06	69	1.41e-04	101	1.33e-04	101
200	10	8.88e-06	27	9.14e-06	56	9.15e-06	55
500	10	9.67e-06	43	9.29e-06	72	8.96e-06	71
500	15	8.93e-06	31	9.87e-06	50	9.06e-06	49
1000	10	9.21e-06	71	9.23e-06	119	9.52e-06	117
1500	10	9.92e-06	95	9.67e-06	166	9.41e-06	165

Table 5. Comparison results of PCM, L-ALM, C-PPA ($OS = 10$).

Problems		PCM		L-ALM		C-PPA	
n	r_a	Rel.err.	Iter.	Rel.err.	Iter.	Rel.err.	Iter.
100	5	8.67e-06	14	7.66e-06	29	7.67e-06	28
100	10	8.39e-06	13	9.10e-06	27	9.12e-06	26
200	5	6.84e-06	21	8.85e-06	36	8.67e-06	35
200	10	5.78e-06	7	6.17e-06	13	8.32e-06	10
500	10	9.13e-06	24	8.22e-06	39	8.86e-06	38
500	15	8.34e-06	17	9.13e-06	26	9.74e-06	25
1000	10	8.22e-06	48	9.25e-06	77	9.33e-06	76
1500	10	9.41e-06	66	9.39e-06	112	9.40e-06	111

Table 6. Comparison results of PCM, L-ALM, C-PPA ($OS = 15$).

Problems		PCM		L-ALM		C-PPA	
n	r_a	Rel.err.	Iter.	Rel.err.	Iter.	Rel.err.	Iter.
100	5	9.62e-06	11	8.92e-06	23	8.99e-06	22
100	10	9.62e-06	13	9.15e-06	27	9.20e-06	26
200	5	4.73e-06	13	9.27e-06	21	8.34e-06	20
200	10	7.98e-06	6	6.15e-06	13	8.34e-06	10
500	10	8.15e-06	17	8.51e-06	25	9.93e-06	24
500	15	4.58e-06	10	4.22e-06	14	7.19e-06	13
1000	10	9.35e-06	35	9.38e-06	56	9.81e-06	55
1500	10	9.75e-06	50	9.20e-06	84	9.46e-06	83

6. Conclusions

This paper proposes a new prediction-correction method for solving the monotone variational inequalities with linear constraints. At the prediction step, the implementation is carried out by a simple projection. At the correction step, the method introduces a simple updating step to generate the new iteration. We establish the global convergence of the method. The convergence condition of the method also allows larger step sizes that can in potential make the algorithm numerically converge faster. The numerical experiments approve the efficiency of the proposed methods. The future work is to explore combining self adaptive technique for the method. Besides, further applications of our method are expected.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by NSF of Shaanxi Province grant number 2020-JQ-485.

Conflict of interest

The authors declare no conflict of interest.

References

1. R. Rockafellar, J. Sun, Solving monotone stochastic variational inequalities and complementarity problems by progressive hedging, *Math. Program.*, **174** (2019), 453–471. https://doi.org/10.1007/s10107-018-1251-y
2. D. Bertsekas, E. Gafni, Projection methods for variational inequalities with application to the traffic assignment problem, In: *Nondifferential and variational techniques in optimization*, Berlin: Springer, 1982, 39–159. https://doi.org/10.1007/BFb0120965
3. S. Dafermos, Traffic equilibrium and variational inequalities, *Transport. Sci.*, **14** (1980), 42–54. https://doi.org/10.1287/trsc.14.1.42
4. E. Gorbunov, N. Loizou, G. Gidel, Extragradient method: O (1/k) last-iterate convergence for monotone variational inequalities and connections with cocoercivity, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022, 366–402.
5. G. Gu, J. Yang, Tight ergodic sublinear convergence rate of the relaxed proximal point algorithm for monotone variational inequalities, *J. Optim. Theory Appl.*, in press. https://doi.org/10.1007/s10957-022-02058-3
6. B. He, W. Xu, H. Yang, X. Yuan, Solving over-production and supply-guarantee problems in economic equilibria, *Netw. Spat. Econ.*, **11** (2011), 127–138. https://doi.org/10.1007/s11067-008-9095-2
7. Y. Gao, W. Zhang, An alternative extrapolation scheme of PDHGM for saddle point problem with nonlinear function, *Comput. Optim. Appl.*, **85** (2023), 263–291. https://doi.org/10.1007/s10589-023-00453-8
8. X. Zhao, J. Yao, Y. Yao, A nonmonotone gradient method for constrained multiobjective optimization problems, *J. Nonlinear Var. Anal.*, **6** (2022), 693–706. https://doi.org/10.23952/jnva.6.2022.6.07
9. S. Bubeck, Convex optimization: algorithms and complexity, *Found. Trends Mach. Le.*, **8** (2015), 231–357. https://doi.org/10.1561/2200000050
10. T. Chan, S. Esedoglu, F. Park, A. Yip, Total variation image restoration: overview and recent developments, In: *Handbook of mathematical models in computer vision*, Boston: Springer, 2006, 17–31. https://doi.org/10.1007/0-387-28831-7_2
11. G. Kutyniok, Theory and applications of compressed sensing, *GAMM-Mitteilungen*, **36** (2013), 79–101. https://doi.org/abs/10.1002/gamm.201310005

12. E. Ryu, W. Yin, *Large-scale convex optimization: algorithms & analyses via monotone operators*, Cambridge: Cambridge University Press, 2022. https://doi.org/10.1017/9781009160865

13. N. Parikh, S. Boyd, Proximal algorithms, *Foundations and Trends® in Optimization*, **1** (2014), 127–239. https://doi.org/10.1561/2400000003

14. M. Hestenes, Multiplier and gradient methods, *J. Optim. Theory Appl.*, **4** (1969), 303–320. https://doi.org/10.1007/BF00927673

15. M. Powell, A method for nonlinear constraints in minimization problems, In: *Optimization*, New York: Academic Press, 1969, 283–298.

16. B. He, H. Yang, S. Wang, Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities, *J. Optim. Theory Appl.*, **106** (2000), 337–356. https://doi.org/10.1023/A:1004603514434

17. G. Qian, D. Han, L. Xu, H. Yang, Solving nonadditive traffic assignment problems: a self-adaptive projection-auxiliary problem method for variational inequalities, *J. Ind. Manag. Optim.*, **9** (2013), 255–274. https://doi.org/10.3934/jimo.2013.9.255

18. X. Fu, B. He, Self-adaptive projection-based prediction-correction method for constrained variational inequalities, *Front. Math. China*, **5** (2010), 3–21. https://doi.org/10.1007/s11464-009-0045-1

19. X. Fu, A general self-adaptive relaxed-PPA method for convex programming with linear constraints, *Abstr. Appl. Anal.*, **2013** (2013), 1–13. https://doi.org/10.1155/2013/492305

20. F. Ma, M. Ni, W. Tong, X. Wu, Matrix completion via extended linearized augmented Lagrangian method of multipliers, *Proceedings of International Conference on Informative and Cybernetics for Computational Social Systems*, 2015, 45–49. https://doi.org/10.1109/ICCSS.2015.7281147

21. Y. Shen, H. Wang, New augmented Lagrangian-based proximal point algorithm for convex optimization with equality constraints, *J. Optim. Theory Appl.*, **171** (2016), 251–261. https://doi.org/10.1007/s10957-016-0991-1

22. B. He, L. Liao, X. Wang, Proximal-like contraction methods for monotone variational inequalities in a unified framework II: general methods and numerical experiments, *Comput. Optim. Appl.*, **51** (2012), 681–708. https://doi.org/10.1007/s10589-010-9373-z

23. B. He, F. Ma, X. Yuan, Optimal proximal augmented Lagrangian method and its application to full jacobian splitting for multi-block separable convex minimization problems, *IMA J. Numer. Anal.*, **40** (2020), 1188–1216. https://doi.org/10.1093/imanum/dry092

24. G. Gu, B. He, X. Yuan, Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach, *Comput. Optim. Appl.*, **59** (2014), 135–161. https://doi.org/10.1007/s10589-013-9616-x

25. B. He, X. Yuan, W. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, *Comput. Optim. Appl.*, **56** (2013), 559–572. https://doi.org/10.1007/s10589-013-9564-5

26. F. Ma, On relaxation of some customized proximal point algorithms for convex minimization: from variational inequality perspective, *Comput. Optim. Appl.*, **73** (2019), 871–901. https://doi.org/10.1007/s10589-019-00091-z

27. F. Ma, M. Ni, A class of customized proximal point algorithms for linearly constrained convex optimization, *Comput. Appl. Math.*, **37** (2018), 896–911. https://doi.org/10.1007/s40314-016-0371-3

28. B. He, F. Ma, X. Yuan, Optimally linearizing the alternating direction method of multipliers for convex programming, *Comput. Optim. Appl.*, **75** (2020), 361–388. https://doi.org/10.1007/s10589-019-00152-3

29. J. Liu, J. Chen, J. Zheng, X. Zhang, Z. Wan, A new accelerated positive-indefinite proximal ADMM for constrained separable convex optimization problems, *J. Nonlinear Var. Anal.*, **6** (2022), 707–723. https://doi.org/10.23952/jnva.6.2022.6.08

30. B. He, F. Ma, S. Xu, X. Yuan, A generalized primal-dual algorithm with improved convergence condition for saddle point problems, *SIAM J. Imaging Sci.*, **15** (2022), 1157–1183. https://doi.org/10.1137/21M1453463

31. F. Ma, Y. Bi, B. Gao, A prediction-correction-based primal-dual hybrid gradient method for linearly constrained convex minimization, *Numer. Algor.*, **82** (2019), 641–662 https://doi.org/10.1007/s11075-018-0618-8

32. S. Xu, A dual-primal balanced augmented Lagrangian method for linearly constrained convex programming, *J. Appl. Math. Comput.*, **69** (2023), 1015–1035. https://doi.org/10.1007/s12190-022-01779-y

33. E. Blum, W. Oettli, *Mathematische optimierung: grundlagen und verfahren*, Berlin: Springer, 1975.

34. P. Mahey, A. Lenoir, A survey on operator splitting and decomposition of convex programs, *RAIRO-Oper. Res.*, **51** (2017), 17–41. https://doi.org/10.1051/ro/2015065

35. B. He, A uniform framework of contraction methods for convex optimization and monotone variational inequality, *Scientia Sinica Mathematica*, **48** (2018), 255. https://doi.org/10.1360/N012017-00034

36. J. Yang, X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, *Math. Comput.*, **82** (2013), 301–329. https://doi.org/10.1090/S0025-5718-2012-02598-1

37. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Le.*, **3** (2011), 1–122. https://doi.org/10.1561/2200000016

38. G. Ye, Y. Chen, X. Xie, Efficient variable selection in support vector machines via the alternating direction method of multipliers, *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011, 832–840.